

Setting up a TARPEN Node

In July 2015, I wrote about TARPEN (Terrestrial Amateur Radio Packet Network), a packet radio networking movement driven by Tadd Torborg, KA2DEW, presently being experimented with in central North Carolina. This month we're going to walk through the process of building a complete TARPEN node.

We've discussed several times in this column why packet radio networks are being phased out. The short version is that the internet was more user-friendly, and few builders were willing to invest in a 1200-baud data network. Yes, many packet networks still exist and are operating, but these are few and far between, and often quite small. As a data point: The NEDA network stretched from Canada to New Jersey at its peak, with over a hundred nodes.

TARPEN is an effort to build a network. It's a new endeavor, somewhat like past efforts from the 1990s, but using modern and much less expen-

sive components. A key differentiator between the TARPEN and things like APRS or TheNET is that it is expected that every person "getting on" TARPEN is actually going to be building and learning about the components of the network.

Participation is key. The design doesn't really address random stations just showing up, so there's no builder vs. user mentality. It's a purposeful limitation that may eventually be changed, but for the moment it lets hams build and operate a good network of semi-permanent stations. Since the components of the TARPEN network are not unique to TARPEN, there is little risk in building one, and you're bound to learn something in the process.

You don't need to be in range of an existing network, although that helps. You *do* need to be working with two radios, a primary and secondary, if you aren't in range of somebody else who wants to try this. It's sort of like trying to get into amateur television: If you don't know somebody else doing it, it can be lonely.

TARPEN is designed to be as open and transparent as possible, so the participants can learn

* c/o CQ magazine

Email: <N2IRZ@cq-amateur-radio.com>

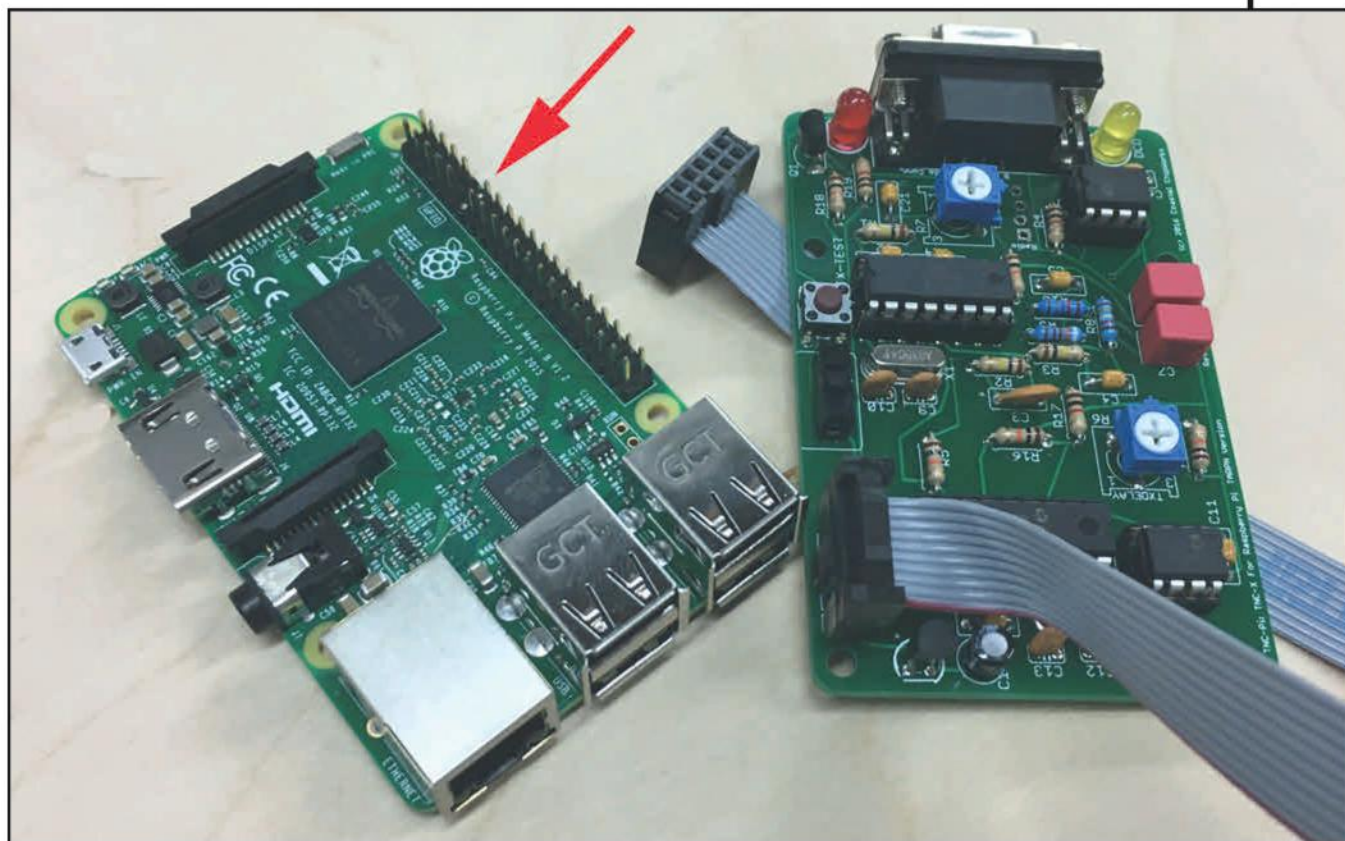


Figure 1. A Raspberry Pi (left) and TNC-Pi, TARPEN version. This is all that's needed to build a TARPEN node, aside from a radio and some wires. Note the cut pins (arrow) on the Raspberry Pi, to accommodate the 10-pin IDC ribbon cable connector. When building the cable, make sure the IDC connectors are oriented the same way, so Pin 1 on the TNC really connects to Pin 1 on the Raspberry Pi.



Figure 2. The TARPNode at NC4FG-2:FIN. The Raspberry Pi (at left) here is supporting three TNC-PIs, connected through a ribbon cable running an I2C Bus. This kind of a cabinet allows for easy setup and troubleshooting.

as much as possible about how networking and data transfer work. I'm not as young as I used to be, but way back when, I learned a lot about networking by building and administering (and writing about) packet networks. This in-depth understanding later helped me at work, where the CAN-Bus networks being used in automobiles were nearly the same as AX.25 networks. I became the go-to guy for CAN-Bus at work, which played a part in my continuing success (I just celebrated my 30th anniversary at work).

A TARPNode can be built and aligned by non-IT hams. It helps to be a little computer-savvy, and to be familiar with packet, but you certainly don't need to be more than a casual user. The resulting network can be used for the fun of transferring data — files, photos, spreadsheets, whatever — and can be pressed into service in an emergency communications situation. Because of how it is built, it won't suffer catastrophic overload (i.e., fail) and temporary nodes can be easily deployed (such as in a strategically-placed vehicle) to handle data transfer needs.

While this column is not nearly large enough to cover all the smallest details of building a TARPNode, we'll go over the basics, and you should visit the TARPNode website <tarpn.net> for the rest. Plan on spending about \$80 for each end of a link, not including radios and antennas, but you probably have some of the smaller items you need on hand. And, of course, hunting for bargains works.

This shopping list is taken from the builder's page on tarpn.net.

- Raspberry Pi microcomputer. Several versions are

acceptable (see the website), list price for the latest Raspberry Pi 3 Model B is \$35.

- TNC-Pi terminal node controller (TNC) kit from Coastal Chipworks <<http://tnc-x.com/TNCPi.htm>>, about \$43 plus shipping. Be sure to get the version optimized for TARPNode, which includes a header and the I2C Data Bus preconfigured, although the original TNC-Pi will work with some effort. Use the link at <<http://bit.ly/2jwgM87>> to get the right one.

- A microSD memory card (for the Pi 3; earlier models use standard SD cards) with at least 8 GB of capacity (about \$5). The TARPNode site recommends an 8-GB Class 4 card, but I used a 16-GB card without any problem. You'll also need something that will let your computer talk to the card (so it can be formatted): I had a small USB converter on hand.

- Power: 12 volts for the radio(s), and USB power (5 volts) for the Raspberry Pi and TNC-PIs. A wall wart for the USB power worked for me, but in actual node service, consider a battery backup system, as described on the TARPNode page.

- Radios, cables, antennas, and other small bits: More on this later.

- A USB mouse and keyboard, HDMI-capable monitor and a computer with an internet connection, for the setup process. All these are only needed temporarily.

The entire process is covered in great detail on the TARPNode site. So much detail, in fact, that it gets out of date, so don't panic if things don't match up perfectly. For the most part, they do, and when they don't, it's minor stuff. Links to the downloads are right there in the instructions, so I strongly recommend following the information there.

While you don't need to be an IT expert, it helps to be comfortable with a computer. Much of the initial setup uses "Command Line" computing, where you have to type in a command to make something happen — no icons or mouse clicks. Users of Linux or DOS know this technique already, but if you aren't sure, just follow the instructions and you should be fine.

Radio Check

The first step is to verify that you're really in simplex range to the other guy. This is as easy as having a voice QSO to verify the path. It can get very frustrating trying to make a link functional if the radio path isn't working.

Raspberry Pi Setup

Next, get a Raspberry Pi and compatible micro (or standard) SD card — depends on which version Pi you bought. Some of the older supported versions can be found at bargain prices: The latest & greatest is not a huge advantage. Format the SD card on your computer using the SD Association Formatter because the format on there when you bought it won't be correct. *Trust me.*

After formatting, download and copy NOOB-LITE to the SD card, then install the card to the Raspberry Pi. Connect a USB keyboard and mouse, Ethernet (from your home router connected to the internet) and HDMI monitor and boot up the Pi (it is powered by a USB connection). Using the on-screen user interface, download and install the Raspbian operating system (a flavor of Debian Linux) — this is nearly 4 GB, so allow some time for this step to finish. Walk away for a while, it runs automatically.

Reboot the Pi (now with Raspbian) and connect to it from your computer using a terminal program. I used PuTTY for Windows, but there are other choices. Type in "sudo raspi-

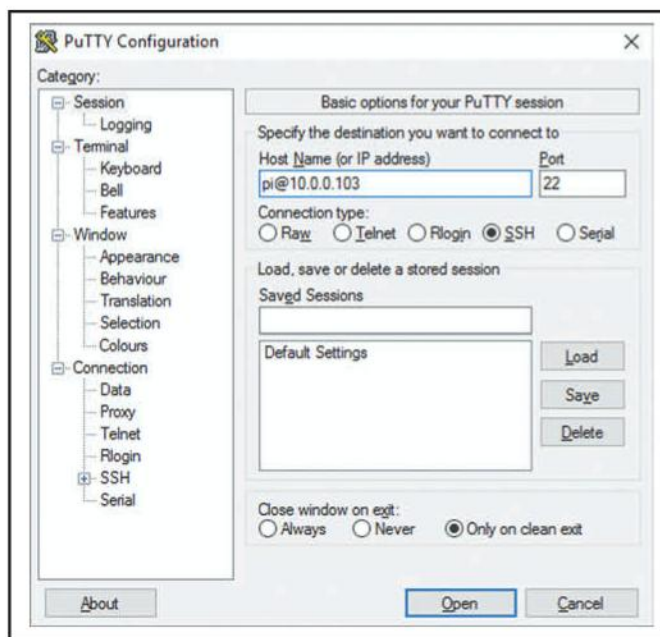


Figure 3. The PuTTY window I use to connect to my Pi, which I have at IP address 10.0.0.103. Note that SSH is selected below. All the default settings are fine. Get PuTTY at <http://www.putty.org/>.

config" to start the configuration routine and configure some basic settings (time zone, node name, I2C Data Bus support, audio output, and disable the serial ports). Then run the TARPEN install/configuration script by copying some text from the website and pasting it into your terminal window. You can retype it, but copying and pasting is much easier. The TARPEN

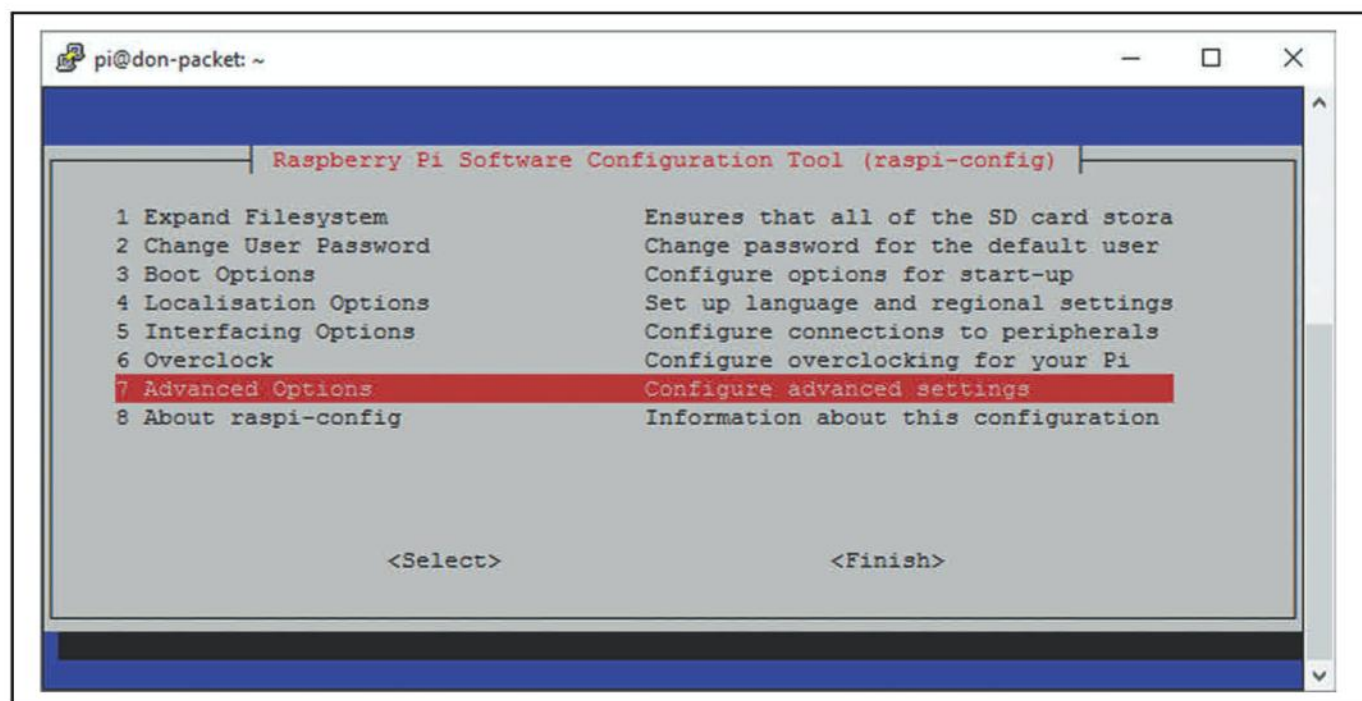


Figure 4. The raspi-config window on the Raspberry Pi, reached by typing "sudo raspi-config" at the Raspbian Linux command prompt. You need to switch SSH mode on here, as well as set a few other parameters. You can get to this window either directly on the Raspberry Pi itself (with a keyboard, mouse, and monitor connected) or through a terminal program on your computer.

```
pi@don-packet: ~  
pi@don-packet:~$ tarpn i2c  
#### -TARPN vJ024 =  
Scanning for attached TNC-Pis  
note: Addresses found are shown in HEXADECIMAL.  
0 1 2 3 4 5 6 7 8 9 a b c d e f  
00: 03  
10:   
20:   
30:   
40:   
50:   
60:   
70:   
pi@don-packet:~$ tarpn i2c-get 3  
#### -TARPN vJ024 =  
Getting TNC-Pi registers at address 3  
Verify proper response by observing the returned I2C Address.  
Inconsistent results may be caused by malfunction in the TNC caused by  
received data arriving from the radio while node software is shut down.  
  
01 TXDelay - Zero means use ADC 0  
02 Persistence 215  
03 Slottime (in 10 mS) 15  
04 TXTail 1  
05 Full Duplex - Not used 0  
06 Our Channel (Hex) 00  
07 I2C Address (0 = async) Hex 03  
ADC Value 30  
8 1 0 d7 f 1 0 0 3 1e cd c0 sum 0  
done  
pi@don-packet:~$
```

Figure 5. From the Raspberry Pi command prompt, we use “tarpn” commands to interact with the TARPN node. This shows only one TNC-Pi recognized, at i2c address 03, and that TNC’s configuration settings. Note that register 7 contains the i2c address for the TNC, which needs to be changed from the default value of 9.

script adds needed, and removes unneeded, software to/from the Raspberry Pi. This cleans up the file structure, and speeds any future updates. It’s doing a lot, so be patient.

My installation ran for about an hour, but your experience may vary. At only one point was my input needed; about halfway through, I had to press “q” to quit an information message. Again, get up and walk away, but check on it from time to time. Oh, and don’t worry about any error messages you see. There will be several (for example “File Not Found!”), but it’s all good.

A Problem!

As usual (for me), I was initially unable to connect to the Pi using PuTTY. Long story short, the Raspbian distribution I installed had “SSH” mode switched off by default, but it is required to be on. The TARPN folks have fixed the website instructions to cover this, but the short answer is, before you can connect from your computer, use RASPI-CONFIG to switch SSH mode on. Save the setting with “Finish” and then connect from your computer.

We use the computer to make configuration easier and somewhat more stable, although you can (if you must) use

the user interface on the Raspberry pi directly for all the setup activities. I’ve found using the computer makes things easier.

Once the TARPN script ends, a few simple tests, detailed on the website, verify all is well.

TNC Setup

The next step is to set up your TNCs. If you haven’t built them yet, do that first. The TARPN version of the TNC-Pi doesn’t include assembly instructions, but there are good ones on the TARPN site. Generally, it should take an hour or so to build. They aren’t all that complex, but take your time and pay particular attention to the orientation of the IC sockets.

TARPN nodes can also be built with old-school TNCs, if you use an RS-232 to USB converter (under \$20) and run the TNC in KISS mode. The webpage has all the details, but since new TNCs are expensive, and not as common at hamfests as they once were, I’m going to stick with the TNC-Pi, which are a lot smaller and convenient.

There are 10 pins on the Raspberry Pi that need to connect to the TNC-Pi, and while this can be done in several ways, I recommend using a 2x5-pin

header with a piece of ribbon cable, as opposed to trying to stack the TNCs atop the Pi. A word of caution: The insulation-displacement headers I bought at Fry’s Electronics are too wide to fit cleanly onto the Raspberry Pi’s header. The TARPN site suggests cutting off the “interfering” pins, so I did.

On your ribbon cable, verify that Pin 1 of the Raspberry Pi really connects to Pin 1 of the TNC-Pi. It is easy to get the connector wrong, with the odd and even pins interchanged. After making this connection, start configuring the TNC-Pi: Connect to the Raspberry Pi as before, and send the command “tarpn i2c,” which returns the address for each i2c device connected (there should be only one, at default address 9).

Read the configuration with the command “tarpn i2c-get”, then use the “tarpn i2c-set” command to set the TNC’s parameters as described in the instructions (Persistence = 215, Slottime = 15, TXTail = 1). Lastly, change the TNC-Pi’s default address from 9 to a unique value (start with 3 and work your way up as you add TNCs) using a command like “tarpn i2c-set 9 7 3” (9 is the original address, register 7 is where the address is stored, and 3 is the new value). Reboot the TNC to see the change.

The last steps before getting on the air are to adjust the transmit audio level (this is really important) and the TXDelay value, as with any packet setup. The instructions on the TARPN site are good, but of course you need both a radio (and power/antenna/etc.) and a cable to connect the radio to the TNC. The TNC end needs a DE-9 male connector, and the pin assignments (which are the same as Kantronics KPC/KAM series TNCs) are as follows:

- Pin 1: TX Data (to radio)
- Pin 3: PTT
- Pin 5: RX Data
- Pin 6: Ground

For the radio’s connections, check your owner’s manual, or the internet. I like the PDF document at <<http://bit.ly/2joas6F>>, courtesy of Buck Rogers, K4ABT, my predecessor on these pages. I’ve found that using the auxiliary data port, if your radio has one, is better than using the microphone connector.

Repeat the TNC process for any other TNCs you plan to use, then string them all together to form a complete node. Yes, you can run with only one TNC, but eventually you’ll want to add links, right?

The last part of the process is configuring and, finally, deploying your node (which runs on the Raspberry Pi). Connect to the Pi and run the “tarprn config” command to set the node parameters: Nodecall (mine is N2IRZ-2), A unique nodename (mine is “don”), the Maidenhead locator of the node, up to 8 lines of an INFO message, a short Connect Text message, and the “local” operator’s callsign (N2IRZ).

Enable ports as needed (one per TNC) and set the port’s I2C address for the TNC-Pi devices. Specify your node’s place in the network by naming your neighbor (the other end of your link). If you don’t (yet) have a neighbor, set it to something interesting, like W1AW-2.

Finally, run the command “tarprn test.” This starts up the BPQ node software and keeps it running until you stop it (or switch off power). Assuming the node started and there were no errors, launch BPQ Terminal on your computer (the TARPn page has download links) and start playing with your node. Check the node with the STATS command, and get a list of all commands by typing a “?” and hitting enter. If you’ve used packet before, these commands should be familiar. If the radio link is working, try CONNECTing to your neighbor.

The very last part is to make permanent connections for the radios, a housing for the entire node, and basically clean things up for permanent installation. If you’re all alone, join the TARPn group at Yahoo and get expert advice on final testing and settings. Otherwise, work with your (experienced) neighbor and get the link tuned in, and start exploring, experimenting, and enjoying your TARPn.

Before I close, please understand that it’s not possible to cover every small nuance of the process in detail. So with that in mind, visit the TARPn website and poke around. When you’re following the TARPn site’s instructions, I advise following them very carefully. I’d have saved myself some grief if I’d done that: The instructions are very detailed, so skimming them over just won’t do.

Make your plans for Dayton/Xenia if you haven’t already: The first Hamvention at the new venue promises to be better than ever. I’ll be at the CQ booth much of the time, so if you want to tell me where to go in person, you know where to find me. Hope to see you there!

– Until next time, 73 de N2IRZ



MFJ Partners With InnovAntennas

MFJ Enterprises and InnovAntennas Limited have announced a partnership which will provide a new and up-to-date range of computer-optimized directional antennas for Hy-Gain and Cushcraft, two of the legacy brands of MFJ Enterprises.

The MFJ facility in Starkville, Mississippi, will be building some of the smaller antennas under the InnovAntennas label using MFJ/Hy-Gain and Cushcraft tooling.

The most popular models will be covered by MFJ, including the LFA (Loop Fed Array) Yagi (pictured above) and OWL (Optimized Wideband Low impedance). Additionally, new computer-optimized and yet unreleased HF and VHF mono and multiband Yagis will be introduced through the Hy-Gain and Cushcraft brands. It is expected that the first in this new line of antennas will be available this spring on a direct and distributor supply basis.

For additional information, Contact <info@innovantennas.com> or <sales@mfjenterprises.com>. Websites: <www.mfjenterprises.com>, <www.innovantennas.com>.

N3FJP Upgrades Amateur Contact Log

N3FJP continues to add new functions to his flagship Amateur Contact Log software. The latest enhancements include:

- Added ability to select the sound card or device (RIGblaster, Signalink, etc.) that you want to use to play your audio .WAV files (Settings > Transmit > Phone).
- Added ability to record .WAV files for transmit (Settings > Transmit > Phone).
- Removed Windows Media Player dependency.
- FlexRadio users can now connect to their rigs via the Flex API to read and change the frequency and mode of the active slice (Settings > Rig Interface and select Flex API).
- Additional DX Spotting filter options, including blocking stations in your country and blocking stations not in the Super Check Partial list (to hopefully cut down on inaccurate DX spots). Click “More Filtering” on the DX Spotting setup form (Super Check Partial filtering for contesting software only).
- The CW% wildcard will now send the county abbreviation instead of the full county name (state QSO party programs only).

These upgrades are free to registered users and are available now. For more information contact: G. Scott Davis, 118 Glenwood Road, Bel Air, MD 21014-5533. Website: <http://www.n3fjp.com>.

New Country List Update from Jim, AD1C

Jim Reisert, AD1C, has released a new country file update for N3FJP software. To install the updated list, just click File > Download Country Files from AD1C. You only have to do that once, from any of the programs, to bring them all up to date. More details on Jim’s latest release are here <www.country-files.com/>.

Note: “What’s New” is not a product review and does not constitute a product endorsement by CQ. Information is primarily provided by manufacturers/vendors and has not necessarily been independently verified.